OpenVAS Vulnerability Assessment and Management Project

Table of Contents

Project Overview

Project Name: OpenVAS Vulnerability Assessment and Security Audit

Purpose: This project demonstrates the complete implementation of OpenVAS (Open Vulnerability Assessment System) for network vulnerability scanning, assessment, and remediation planning in a controlled virtual environment.

Environment: Kali Linux running on Oracle VirtualBox

Scope: Local network vulnerability assessment, security posture evaluation, and compliance verification

What is OpenVAS?

Definition

OpenVAS (Open Vulnerability Assessment Scanner) is a comprehensive open-source vulnerability scanning and management solution. It is part of the Greenbone Vulnerability Management (GVM) framework and provides capabilities for:

- Network vulnerability detection
- Security configuration assessment
- Compliance auditing
- Continuous security monitoring
- Automated vulnerability testing

Key Components

1. **GVM Scanner (OpenVAS)** - The actual scanning engine
2. **GVM Manager (gvmd)** - Central management daemon
3. **Greenbone Security Assistant (GSA)** - Web-based user interface
4. **PostgreSQL Database** - Stores scan results and configurations
5. **Redis** - Cache for NVT (Network Vulnerability Tests) data

How OpenVAS Works

OpenVAS operates by:

1. Performing unauthenticated and authenticated network scans
2. Testing for over 50,000 known vulnerabilities (NVTs)
3. Identifying security misconfigurations
4. Detecting outdated software versions
5. Providing severity ratings and remediation guidance

Prerequisites

Hardware Requirements

- **RAM:** Minimum 4GB (8GB recommended)
- **Disk Space:** Minimum 20GB free space
- **Processor:** 2+ cores recommended
- **Network:** Active network connection

Software Requirements

- Oracle VirtualBox (Latest version)
- Kali Linux ISO (Latest version)
- At least one target machine for scanning (can be another VM)

Knowledge Prerequisites

- Basic Linux command line operations
- Understanding of networking concepts (IP addresses, ports, protocols)
- Basic cybersecurity concepts
- Virtual machine management

Installation Guide

Step 1: Set Up Kali Linux on VirtualBox

1. Download Kali Linux ISO from official website
2. Create new VM in VirtualBox:
    - Name: Kali-OpenVAS
    - Type: Linux
    - Version: Debian (64-bit)
    - RAM: 4096 MB minimum
    - Storage: 40GB dynamically allocated

3. Configure VM Settings:
    - Network: NAT or Bridged Adapter
    - Enable PAE/NX
    - Allocate 2+ CPU cores
4. Install Kali Linux following the standard installation process

Step 2: Update Kali Linux System

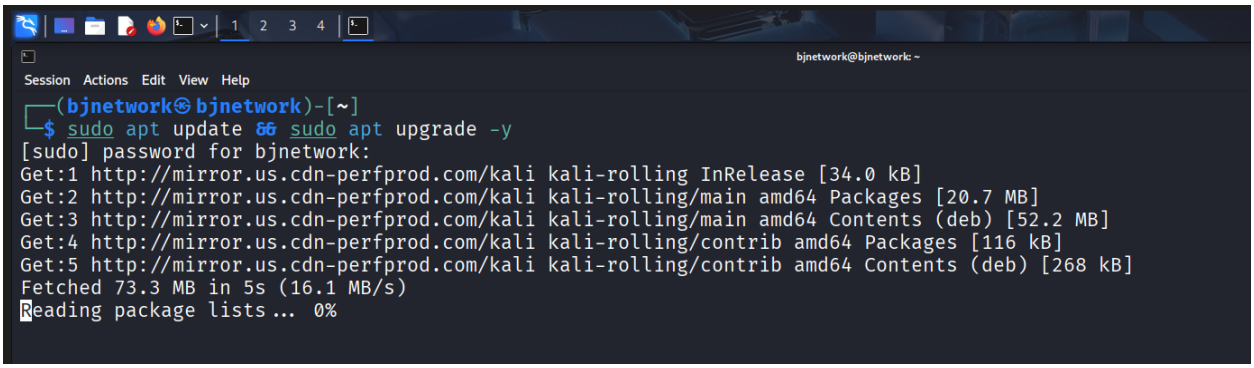
Open terminal and run:

bash

sudo apt update && sudo apt upgrade -y

sudo apt dist-upgrade -y

sudo reboot



Step 3: Install OpenVAS (GVM)

bash

Install GVM (Greenbone Vulnerability Management)

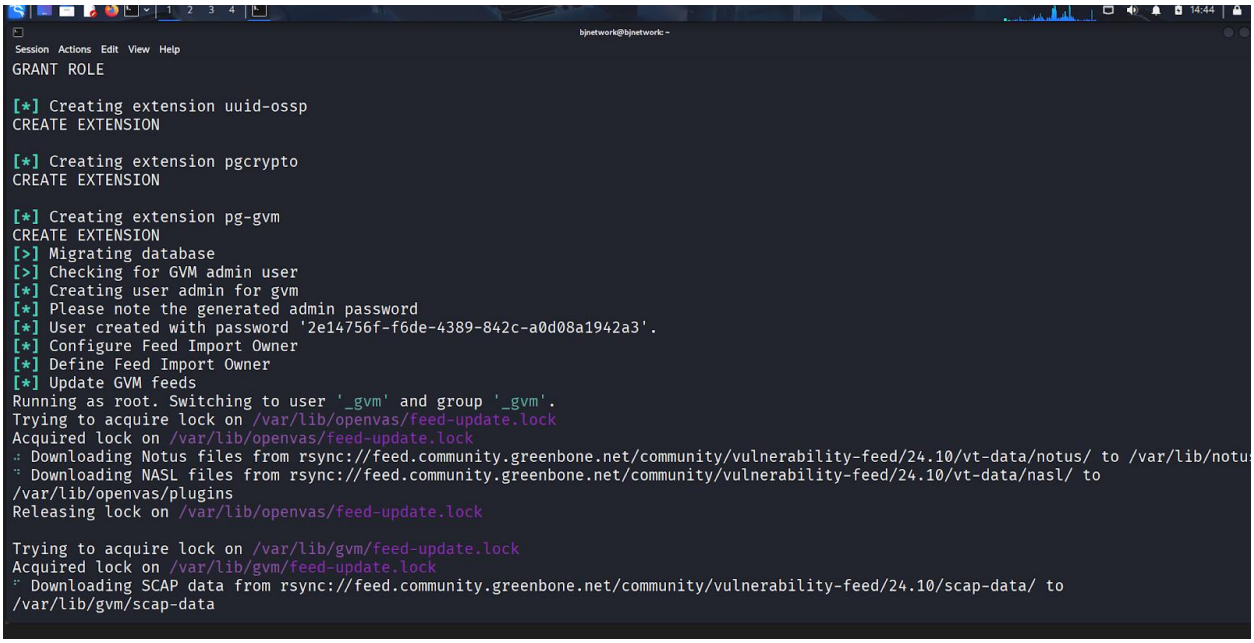
sudo apt install gvm -y

Step 4: Run Initial Setup

bash

Run the setup script

sudo gvm-setup



This process may take 15-30 minutes. The script will:

- Configure PostgreSQL database
- Download and update vulnerability feeds (NVTs)
- Create certificates
- Configure scanner and manager
- Set up the admin user

IMPORTANT: Save the admin password displayed at the end of setup!

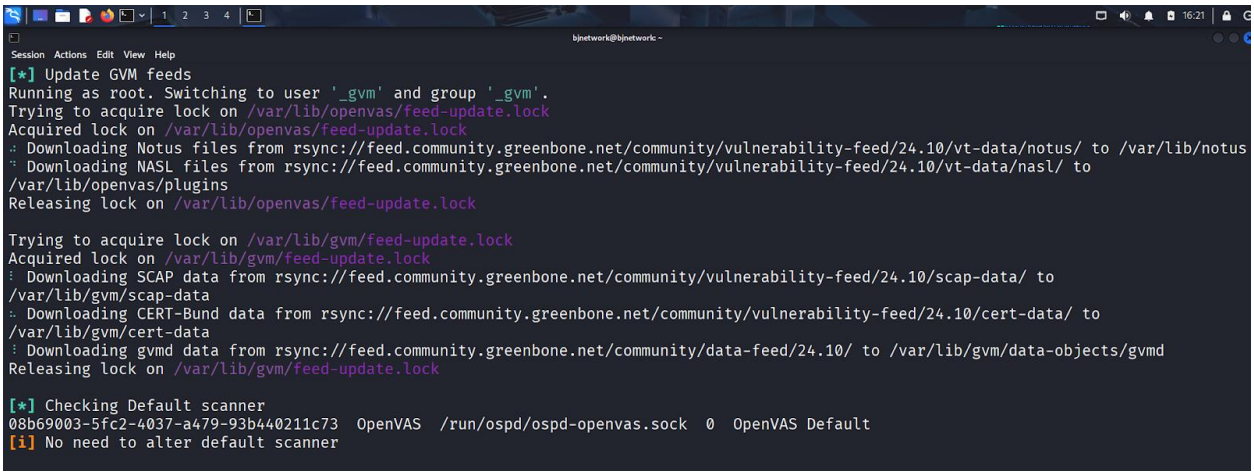
Step 5: Verify Installation

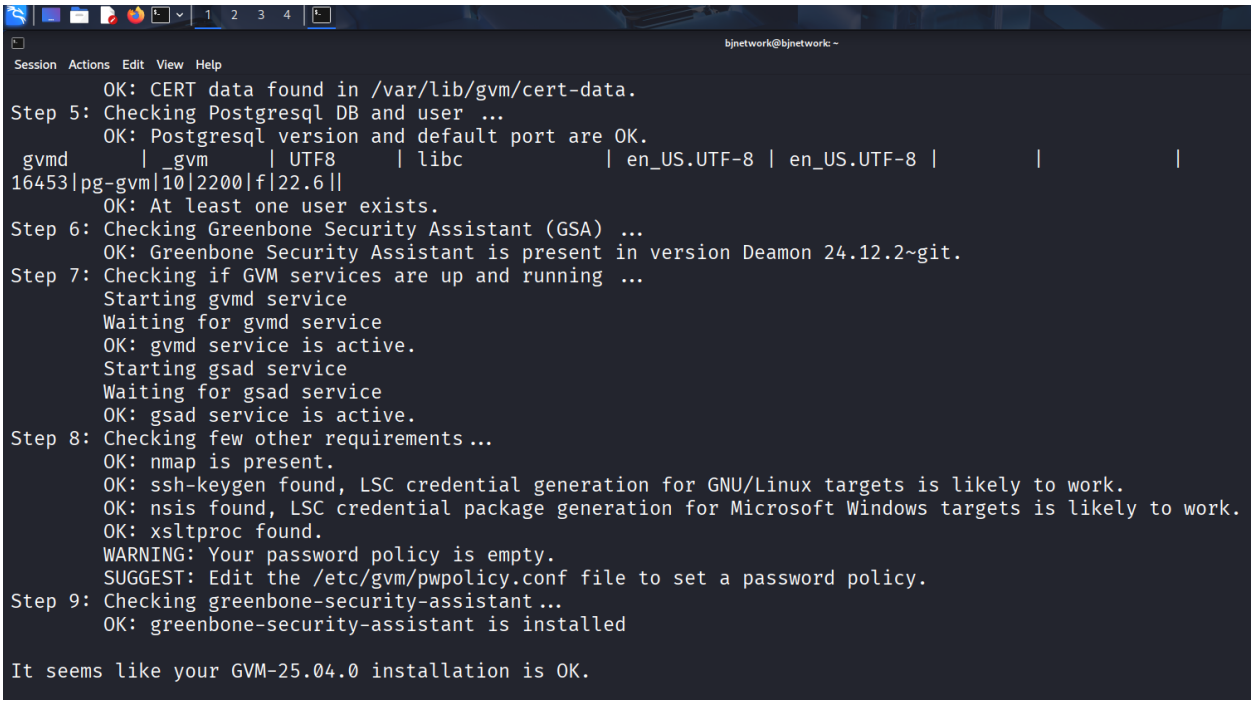
bash

Check GVM services status

sudo gvm-check-setup

Expected output should show all checks passed.





Configuration and Setup

Step 1: Start GVM Services

bash

Start all GVM services

sudo gvm-start

Verify services are running

sudo systemctl status ospd-openvas

sudo systemctl status gvmd

sudo systemctl status gsad

```
┌──(bjnetwork㉿bjnetwork)-[~]
└─$ sudo gvm-start
[i] GVM services are already running

┌──(bjnetwork㉿bjnetwork)-[~]
└─$ sudo systemctl status gvmd
● gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)
     Loaded: loaded (/usr/lib/systemd/system/gvmd.service; disabled; preset: disabled)
     Active: active (running) since Thu 2026-01-22 16:24:09 EST; 3min 28s ago
 Invocation: 4fa7165d1f7a4b438dd537937269d4bc
       Docs: man:gvmd(8)
    Process: 12385 ExecStart=/usr/sbin/gvmd --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm (code=exited, status=0/SUC>
   Main PID: 12387 (gvmd)
      Tasks: 4 (limit: 4657)
     Memory: 464.4M (peak: 465.5M)
        CPU: 1min 1.466s
     CGroup: /system.slice/gvmd.service
             ├─12387 "gvmd: Waiting " --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm
             ├─12406 gpg-agent --homedir /var/lib/gvm/gvmd/gnupg --use-standard-socket --daemon
             ├─12423 "gvmd: Synchron" --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm
             └─12428 "gvmd: Syncing " --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm

Jan 22 16:24:08 bjnetwork systemd[1]: Starting gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)...
Jan 22 16:24:08 bjnetwork systemd[1]: gvmd.service: Can't open PID file '/run/gvmd/gvmd.pid' (yet?) after start: No such file or direc>
Jan 22 16:24:09 bjnetwork systemd[1]: Started gvmd.service - Greenbone Vulnerability Manager daemon (gvmd).
lines 1-19/19 (END)
```

```
┌──(bjnetwork㉿bjnetwork)-[~]
└─$ sudo greenbone-feed-sync --type GVMD_DATA
[sudo] password for bjnetwork:
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
Trying to acquire lock on /var/lib/gvm/feed-update.lock
/var/lib/gvm/feed-update.lock is locked by another process. Waiting 5 seconds before next try.
```

Step 2: Update Vulnerability Feeds

bash

Update NVT feeds

sudo greenbone-feed-sync --type GVMD_DATA

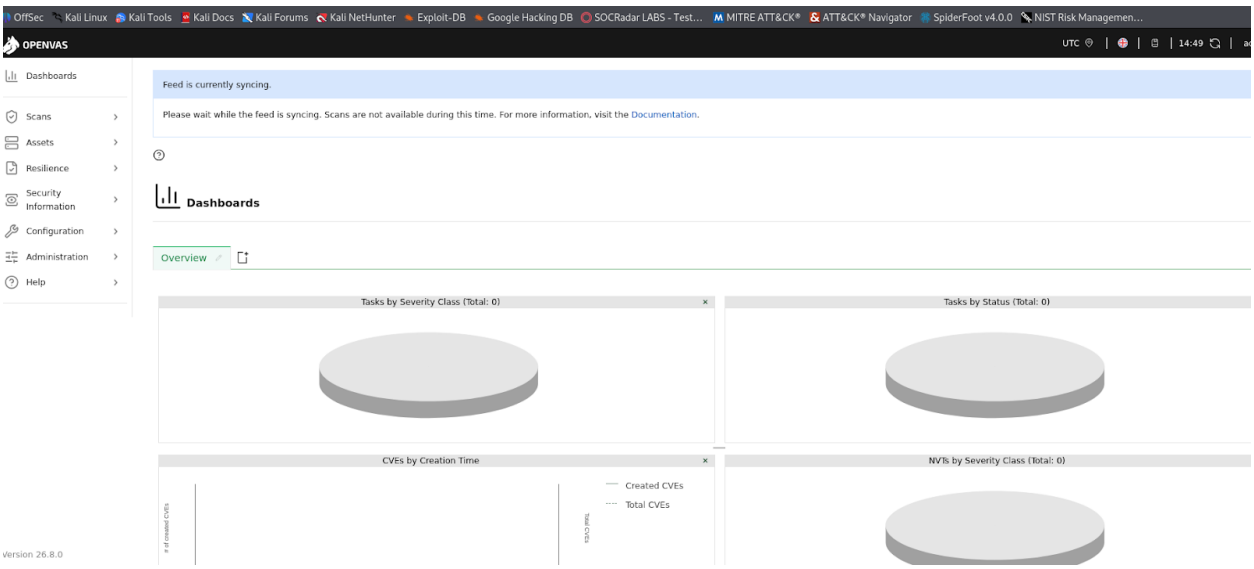
sudo greenbone-feed-sync --type SCAP

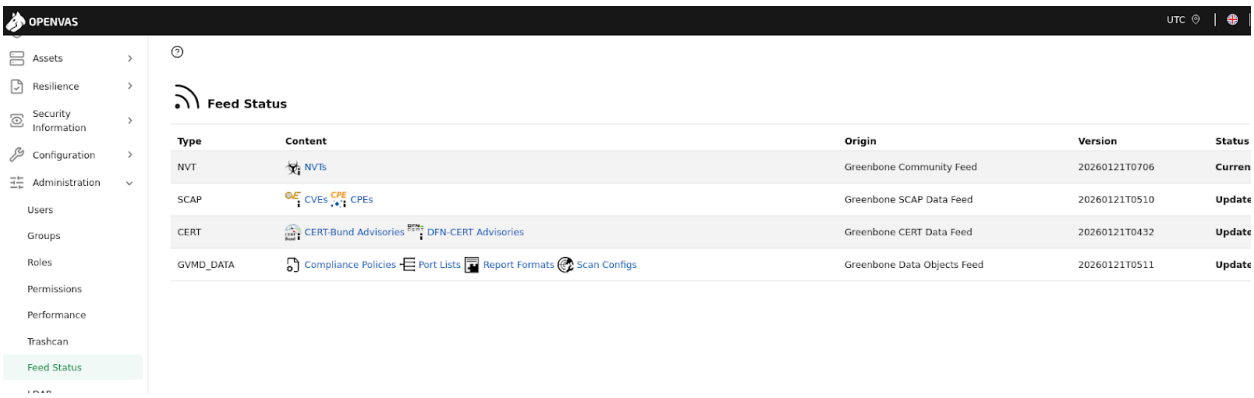
sudo greenbone-feed-sync --type CERT

Alternative single command

sudo runuser -u _gvm -- greenbone-nvt-syncsodu

This may take 1-2 hours for first-time updates.





Step 3: Access Web Interface

1. Open Firefox browser in Kali Linux
2. Navigate to: https://127.0.0.1:9392 or https://localhost:9392
3. Accept the security certificate warning
4. Login with:
    - Username: admin
    - Password: (from gvm-setup output)

Step 4: Create Additional User (Optional)

bash

Create a new user

sudo runuser -u _gvm -- gvmd --create-user=scanner_user --password=SecurePass123

Modify existing user password

sudo runuser -u _gvm -- gvmd --user=admin --new-password=NewSecurePass123

Vulnerability Scanning Process

Step 1: Define Target

Navigate: Configuration → Targets → New Target

Configuration:

- Name: Lab Network Scan
- Hosts: Manual entry or from file
    - Single IP: 192.168.1.100
    - Range: 192.168.1.1-254
    - CIDR: 192.168.1.0/24
- Port List: All IANA assigned TCP and UDP

Screenshot Location: screenshots/11_target_creation/

Step 2: Configure Scan Task

Navigate: Scans → Tasks → New Task

Configuration:

- Name: Network Security Assessment
- Scan Targets: Select created target
- Scanner: OpenVAS Default
- Scan Config: Choose based on scan type:
    - **Full and Fast:** Comprehensive, faster scan
    - **Full and Deep:** Most thorough, slower
    - **System Discovery:** Quick host discovery
    - **Full and Fast Ultimate:** Maximum coverage

Advanced Options:

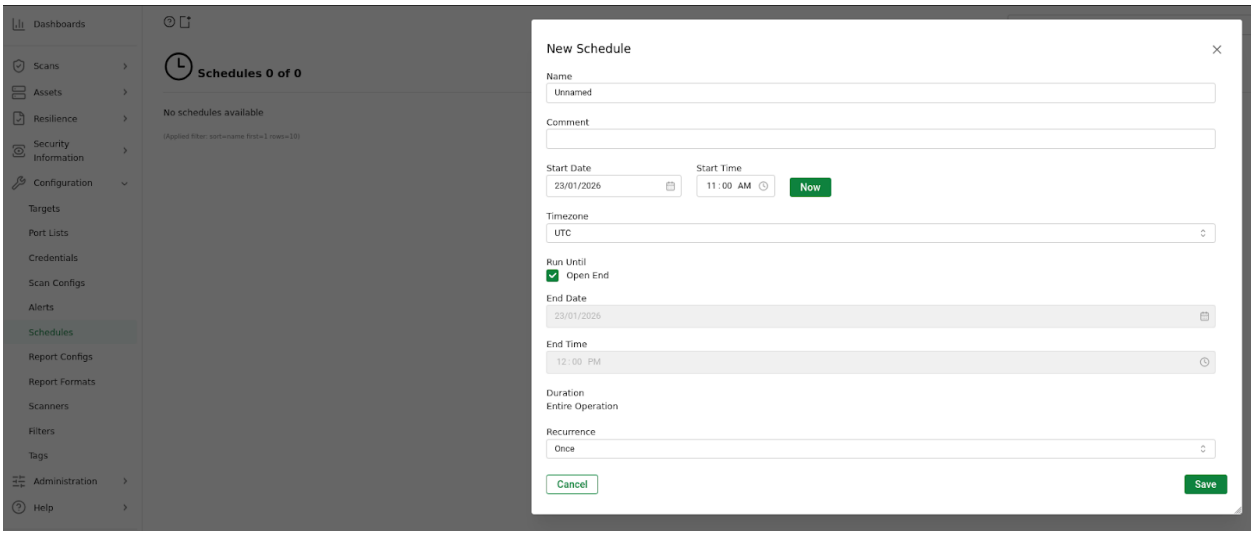
- Maximum Concurrency: 4-10 (adjust based on resources)
- Network Source Interface: Auto
- Order for target hosts: Sequential or Random

Step 3: Create Scan Schedule (Optional)

Navigate: Configuration → Schedules → New Schedule

Configuration:

- Name: Weekly Security Scan
- First Time: Select date/time
- Period: Weekly
- Duration: 2 hours



Step 4: Launch Scan

1. Navigate to: Scans → Tasks
2. Select your task
3. Click the play button (▶) to start scan
4. Monitor progress in real-time

Important Commands During Scanning

bash

Monitor system resources

htop

Check scanner process

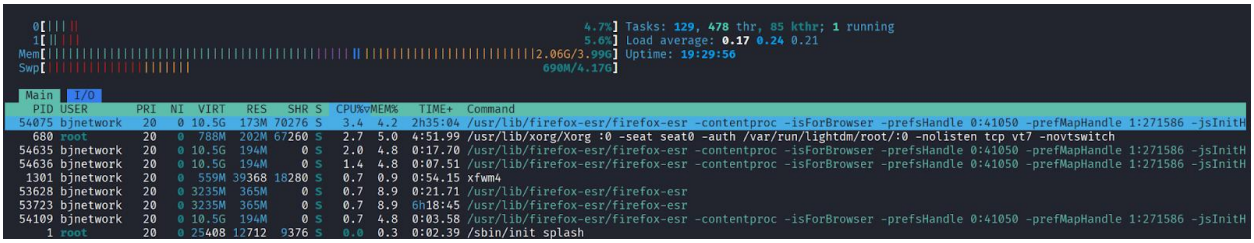
ps aux | grep openvas

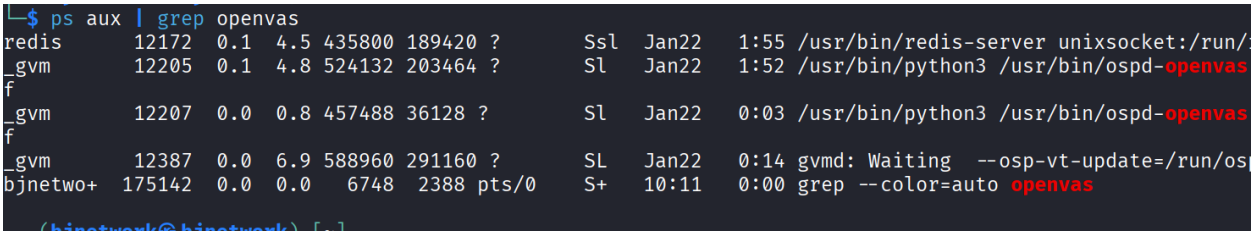
View scanner logs

sudo tail -f /var/log/gvm/openvas.log

Check database activity

sudo -u postgres psql gvmd -c "SELECT COUNT(*) FROM results;"





Results Analysis

Step 1: View Scan Results

Navigate: Scans → Reports → Select completed scan

Report Dashboard Elements

1. **Severity Distribution Graph**
    - High (Red): Critical vulnerabilities requiring immediate action
    - Medium (Orange): Significant security issues
    - Low (Yellow): Minor security concerns
    - Log (Blue): Informational findings

2. **Vulnerability Count Summary**
    - Total vulnerabilities found
    - Breakdown by severity
    - False positive indicators
3. **Host Overview**
    - All scanned hosts
    - Vulnerabilities per host
    - Service detection results

Step 2: Detailed Vulnerability Analysis

Click on any vulnerability to view:

Key Information Fields:

1. **NVT Details**
    - NVT Name
    - OID (Object Identifier)
    - CVE ID (if applicable)
    - CVSS Score

2. **Summary**
    - Clear description of vulnerability
    - Affected component
    - Attack vector
3. **Impact**
    - Confidentiality impact
    - Integrity impact
    - Availability impact

4. **Solution**
    - Remediation steps
    - Patches or workarounds
    - Configuration changes
5. **References**
    - CVE links
    - Vendor advisories
    - Security bulletins

Step 3: Filter and Sort Results

Filtering Options:

- By Severity: High, Medium, Low, Log
- By Host: Specific IP addresses
- By Port: Specific services
- By NVT: Specific vulnerability types

bash

Export results via CLI

sudo -u _gvm gvm-cli socket --xml "<get_reports report_id='REPORT_ID'/>"

Critical Information to Look For

1. High and Critical Vulnerabilities

What to Look For:

- CVSS Score 7.0 or higher
- Remote Code Execution (RCE) vulnerabilities
- Authentication bypass issues
- SQL Injection vulnerabilities
- Cross-Site Scripting (XSS) in web applications
- Privilege escalation vulnerabilities

Example Critical Findings:

- EternalBlue (MS17-010)
- BlueKeep (CVE-2019-0708)
- Log4Shell (CVE-2021-44228)
- ProxyShell Exchange vulnerabilities

2. Missing Security Patches

What to Look For:

- Outdated operating system versions
- Unpatched software applications
- End-of-Life (EOL) software
- Missing security updates

How to Identify:

- Check "Version Detection" results
- Compare against vendor security bulletins
- Review "Product Detection" findings